Name: Dhwani Ravi Panjwani
Andrew ID: dpanjwan

PROJECT – 1

TASK 1

a) Screen shots of input, MD5 and SHA-256 output, both in hex and base 64

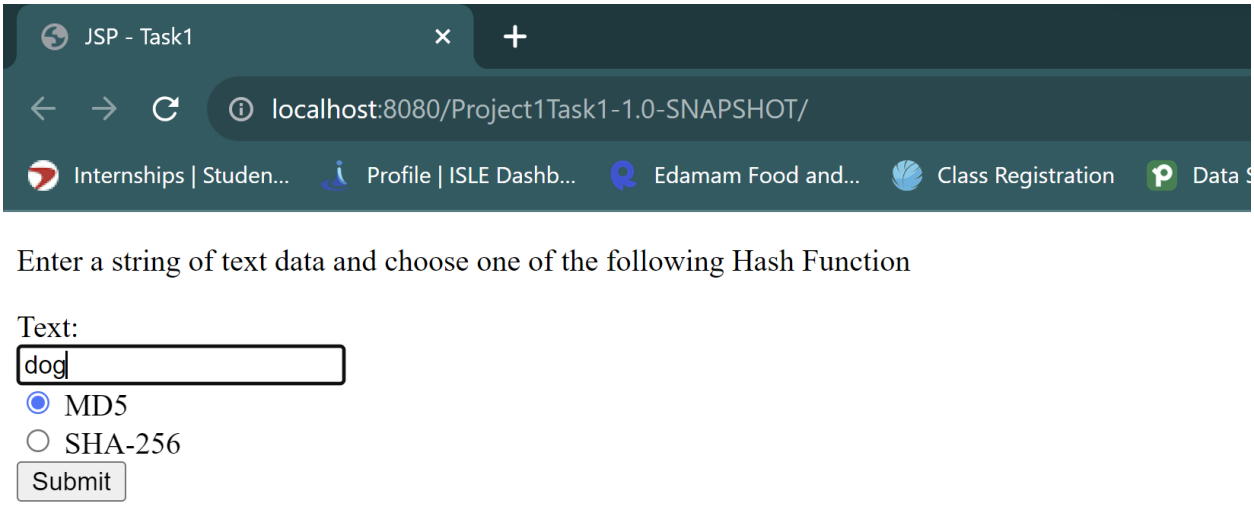
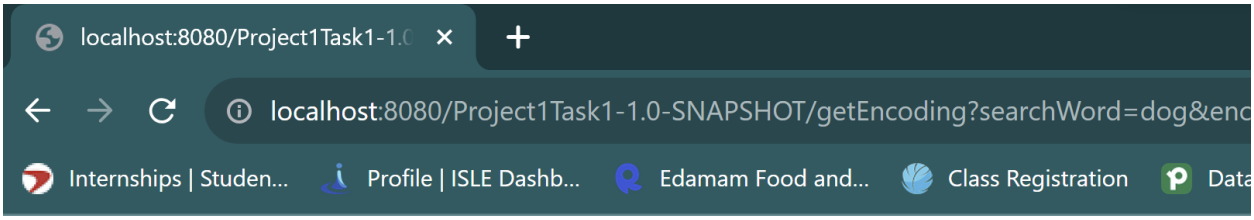


Fig 1 : Entering the string in the text box and choosing the option (MD5 – default)

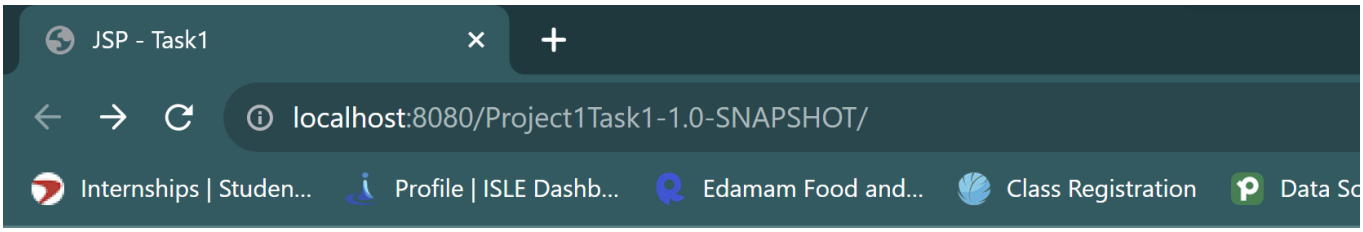


Conversions

MD5 base64 conversion BtgOsMULSaUJtJ8kJOjIBQ==

MD5 hex conversion 06D80EB0C50B49A509B49F2424E8C805

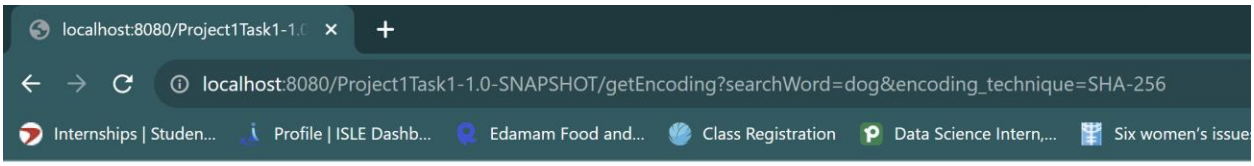
Fig 2: MD5 base64 and hex conversions



Enter a string of text data and choose one of the following Hash Function

Text:
dog
MD5
SHA-256
Submit

Fig 3: Entering the string dog in the text box and choosing the SHA-256 option



Conversions

SHA-256 base64 conversion zWNX792WbejAyy+HbMiex0zjXwlo4RdDmHCEvUL7iUQ=

SHA-256 hex conversion CD6357EFDD966DE8C0CB2F876CC89EC74CE35F0968E11743987084BD42FB8944

Fig 4: SHA-256 base 64 and hex conversions

b) Code snippets for computation of each Hash

```
// Function Reference: Lab 1 code - Hello World Lab 1 code and
Interesting picture Lab 2 code
public void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException {
    // Set the content type of the HTTP response to HTML
    response.setContentType("text/html");

    // Get the 'searchWord' and 'encoding_technique' parameters from
the HTTP request
```

```
    String search = request.getParameter("searchWord");
    String encodingAlgorithm =
request.getParameter("encoding_technique");

    try {
        MessageDigest md = null;
        String algorithm = null;

        // Check if the selected encoding algorithm is MD5
        if (Objects.equals(encodingAlgorithm, "MD5")) {
            md = MessageDigest.getInstance("MD5");
            algorithm = "MD5";
        } else if (Objects.equals(encodingAlgorithm, "SHA-256")) {
            // Check if the selected encoding algorithm is SHA-256
            md = MessageDigest.getInstance("SHA-256");
            algorithm = "SHA-256";
        }

        // Compute the digest of the 'search' string using the selected
algorithm
        md.update(search.getBytes());
        String base64 =
DatatypeConverter.printBase64Binary(md.digest());
        md.update(search.getBytes());
        String hex = DatatypeConverter.printHexBinary(md.digest());

        // Output the computed digest values to the console
        System.out.println(base64);
        System.out.println(hex);

        // Get a PrintWriter from the response object to send an HTML
document back to the caller
        PrintWriter out = response.getWriter();

        // Send an HTML response to the caller
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");

        // Display the computed digest values in base64 and hex formats
        out.println("<p>" + algorithm + " base64 conversion " +
base64);
        out.println("<p>" + algorithm + " hex conversion " + hex);

        out.println("</body></html>");
    } catch (NoSuchAlgorithmException e) {
        // Handle the case where the selected encoding algorithm is not
available
        System.out.println("No encoding available" + e);
    }
}
```

TASK – 2

a) Screen shots of input page(s) and output page(s).

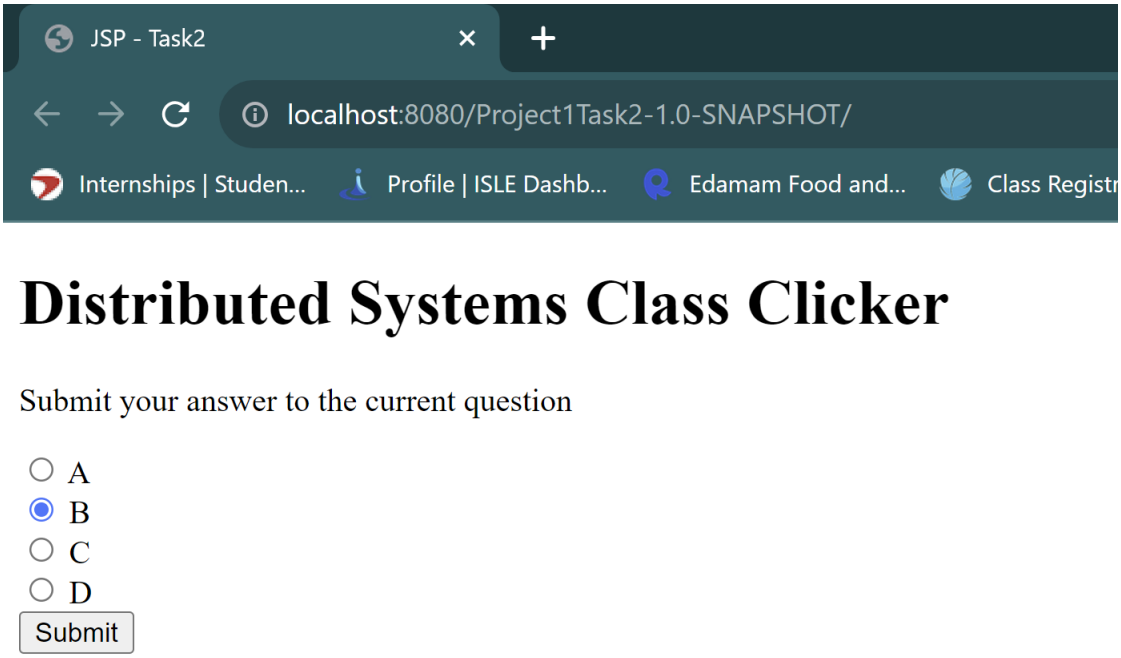
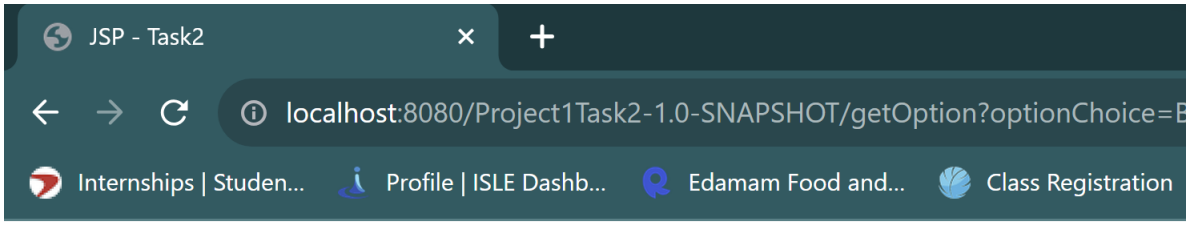


Fig 1: Input Page with Option B selected



Distributed Systems Class Clicker

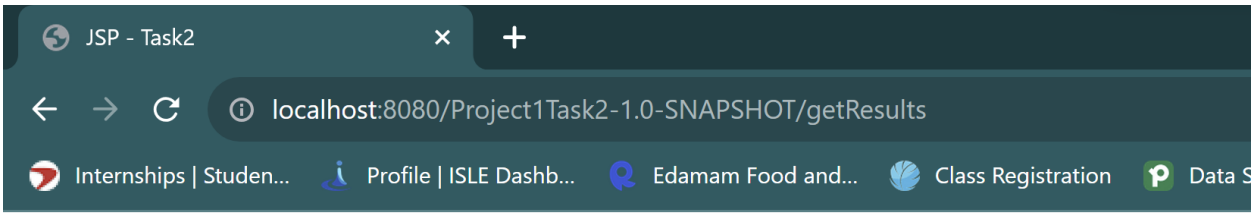
Your B has been registered

Submit your answer to the current question

- A
- B
- C
- D

Submit

Fig 2: Input Page 2 which shows that B has been registered



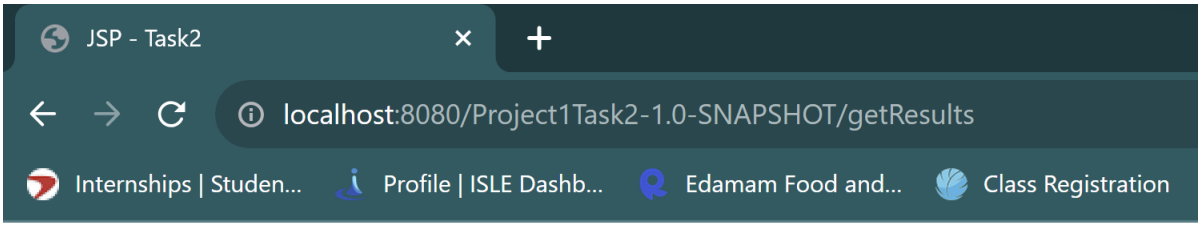
Distributed Systems Class Clicker

B: 2

C: 1

D: 1

Fig 3: This shows the output page getResults which shows the number of times each alphabet was selected



Distributed Systems Class Clicker

There are currently no results

Fig 4: On refreshing the page, we get no results found

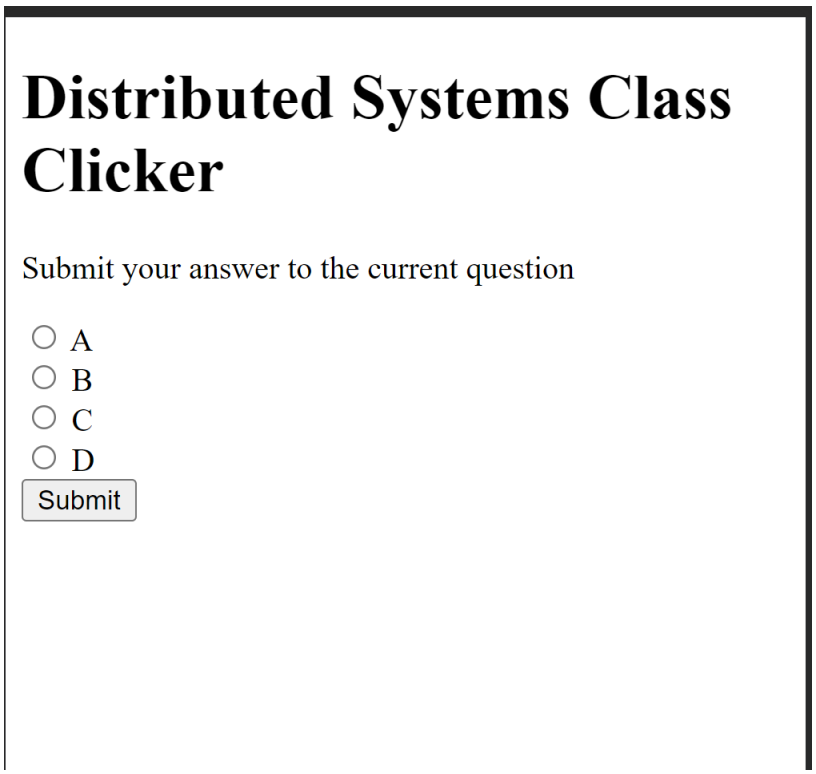


Fig 5: Mobile view Get Options

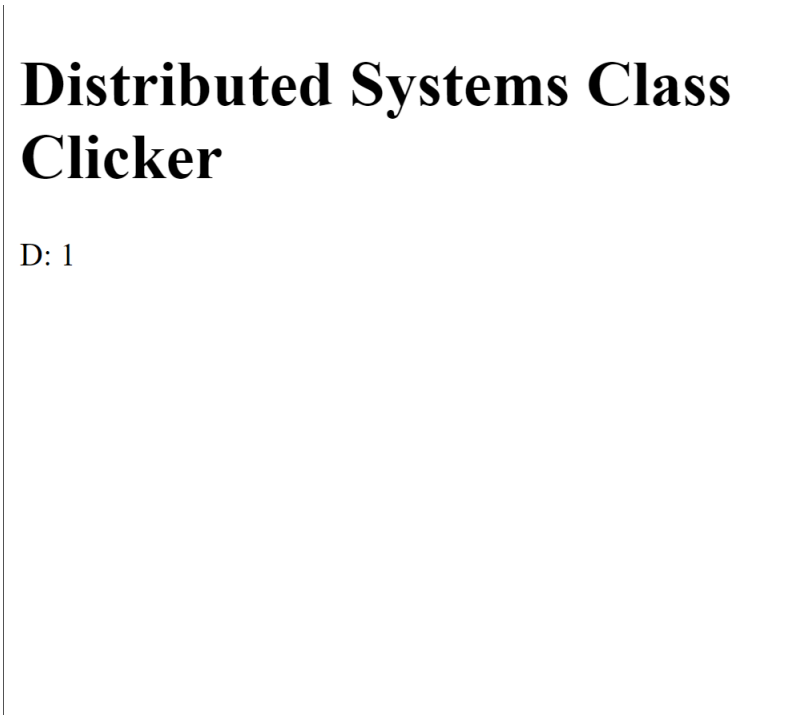


Fig 6: Mobile view Get Results

b) Code snippets for producing clicker output.
Servlet:

c)
```
@WebServlet(name = "choosingOptionServlet", value = {"/getOption",
"/getResults"})
public class PollClickerServlet extends HttpServlet {


    PollClickerModel pcm = null;  // The "business model" for this
app

    //Initializing the hashmap which will be used to store the count
results
    HashMap<Character, Integer> scoreCount = new HashMap<>();

    //Initializing the model
    public void init() {pcm = new PollClickerModel();
    }

    // This servlet will reply to HTTP GET requests via this doGet
method
    //Reference for the doGet function: Interesting Picture: Lab 2
and Lab 1
    protected void doGet(HttpServletRequest request,
                         HttpServletResponse response)
            throws ServletException, IOException {
        String option = request.getParameter("optionChoice");

        String ua = request.getHeader("User-Agent");

        boolean mobile;
        // prepare the appropriate DOCTYPE for the view pages
        if (ua != null && ((ua.indexOf("Android") != -1) ||
(ua.indexOf("iPhone") != -1))) {
            mobile = true;
            /*
             * This is the latest XHTML Mobile doctype. To see the
difference it
             * makes, comment it out so that a default desktop
doctype is used
             * and view on an Android or iPhone.
             */
            request.setAttribute("doctype", "<!DOCTYPE html PUBLIC
\"-//WAPFORUM//DTD XHTML Mobile 1.2//EN\"
\"http://www.openmobilealliance.org/tech/DTD/xhtml-
mobile12.dtd\">");
        } else {
            mobile = false;
            request.setAttribute("doctype", "<!DOCTYPE HTML PUBLIC
\"-//W3C//DTD HTML 4.01 Transitional//EN\"
\"http://www.w3.org/TR/html4/loose.dtd\">");
        }

        String nextView;

        //checking if the path contains /getResults
        if (!request.getServletPath().equals("/getResults")) {
            request.setAttribute("getTheOptionChoice", option);
```

```
            if(option!=null) {
                //storing the returned hashmap from model in
scoreCount
                scoreCount = pcm.keepScoreCount(option);
            }

            //setting the attribute getTheScoreCount to scoreCount
            request.setAttribute("getTheScoreCount", scoreCount);

            //forwarding to the next view to fetch more inputs from
the user
            nextView = "index.jsp";
            RequestDispatcher view =
request.getRequestDispatcher(nextView);
            view.forward(request, response);

            //if url path does not contain getResults
        } else {
            if (scoreCount.isEmpty()) {
                //flag for keeping track of no results found or
refreshing the page
                int flag = 1;
                request.setAttribute("flag",flag );

                //forwarding to the next view
                nextView = "result.jsp";
                RequestDispatcher view =
request.getRequestDispatcher(nextView);
                view.forward(request, response);
            } else {
                int flag = 0;


                request.setAttribute("flag",flag);

                //setting the attributes for ACount, BCount, CCount
and DCount
                for (char key : scoreCount.keySet()) {
                    int value = scoreCount.get(key);
                    if(key=='A')
                        request.setAttribute("ACount", value);
                    if (key=='B')
                        request.setAttribute("BCount", value);
                    if (key=='C')
                        request.setAttribute("CCount", value);
                    if (key=='D')
                        request.setAttribute("DCount", value);


                }
                request.setAttribute("getTheScoreCount",
scoreCount);

                //forwarding to the next view
                nextView = "result.jsp";
                RequestDispatcher view =
request.getRequestDispatcher(nextView);
```

```
                    view.forward(request, response);
                    scoreCount.clear();
                    request.setAttribute("getTheScoreCount",
    scoreCount);

                }
            }



            }
```

Model:

```
HashMap<Character, Integer> scoreCount = new HashMap<Character, Integer>();

// Define a method named keepScoreCount that takes a String parameter named
'option'
// and may throw an UnsupportedEncodingException

//Syntax Reference: https://www.geeksforgeeks.org/java-util-hashmap-in-java-
with-examples/
public HashMap keepScoreCount(String option) throws
UnsupportedEncodingException {
    // Check if the HashMap does not contain the Character at the beginning
of the 'option' string
    if (!(scoreCount.containsKey(option.charAt(0)))) {
        // If not, add the Character to the HashMap with a value of 1
        scoreCount.put(option.charAt(0), 1);
    } else {
        // If the Character is already in the HashMap, retrieve its current
count
        int count = scoreCount.get(option.charAt(0));
        // Increment the count by 1
        count = count + 1;
        // Update the HashMap with the new count
        scoreCount.put(option.charAt(0), count);
    }
    // Print the current state of the scoreCount HashMap to the console
    System.out.println(scoreCount);
    // Return the updated scoreCount HashMap
    return scoreCount;
}
```

Index.jsp

```
<% if (request.getAttribute("doctype") != null) { %>
<%= request.getAttribute("doctype") %>
<% } %>

<!DOCTYPE html>
<html>
<head>
    <title>JSP - Task2</title>
```

```
</head>
<body>
<h1><%= "Distributed Systems Class Clicker" %></h1>
<% if (request.getAttribute("getTheOptionChoice") != null) { %>
<p>Your <%= request.getAttribute("getTheOptionChoice") %> has been
registered</p>
<% } %>

<p><%= "Submit your answer to the current question" %>
</p>
<form action="getOption" method="GET">

    <input type="radio" id="a" name="optionChoice" value="A">
    <label for="a">A</label><br>
    <input type="radio" id="b" name="optionChoice" value="B">
    <label for="b">B</label><br>
    <input type="radio" id="c" name="optionChoice" value="C">
    <label for="c">C</label><br>
    <input type="radio" id="d" name="optionChoice" value="D">
    <label for="d">D</label><br>
    <input type="submit" value="Submit">
    <!Reference: https://www.w3schools.com/html/html_forms.asp>
    <!Reference: Interesting Picture - Lab 2 (result.jsp)>
</form>
<br/>

</body>
</html>
```

Result.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<% if (request.getAttribute("doctype") != null) { %>
<%= request.getAttribute("doctype") %>
<% } %>


<!DOCTYPE html>
<html>
<head>
    <title>JSP - Task2</title>
</head>
<body>
<h1><%= "Distributed Systems Class Clicker" %></h1>


<form action="getResults" method="GET">
    <%
        Object flagAttribute = request.getAttribute("flag");
        if (flagAttribute.equals(1)) {
    %>
    <p>There are currently no results</p>
    <%
    } else {
    %>
    <% if (request.getAttribute("ACount") != null) { %>
```

```
    <p>A: <%= request.getAttribute("ACount") %> </p>
    <% } %>
    <% if (request.getAttribute("BCount") != null) { %>
    <p>B: <%= request.getAttribute("BCount") %> </p>
    <% } %>
    <% if (request.getAttribute("CCount") != null) { %>
    <p>C: <%= request.getAttribute("CCount") %> </p>
    <% } %>
    <% if (request.getAttribute("DCount") != null) { %>
    <p>D: <%= request.getAttribute("DCount") %> </p>
    <% } %>
    <%
        }
    %>
    <!Reference: https://www.w3schools.com/html/html_forms.asp>
    <!Reference: Interesting Picture - Lab 2 (result.jsp)>
</form>
<br/>

</body>
</html>
```

TASK 3

a) Screen shots of two uses of the input page (two different sets of input data) and the corresponding output pages.

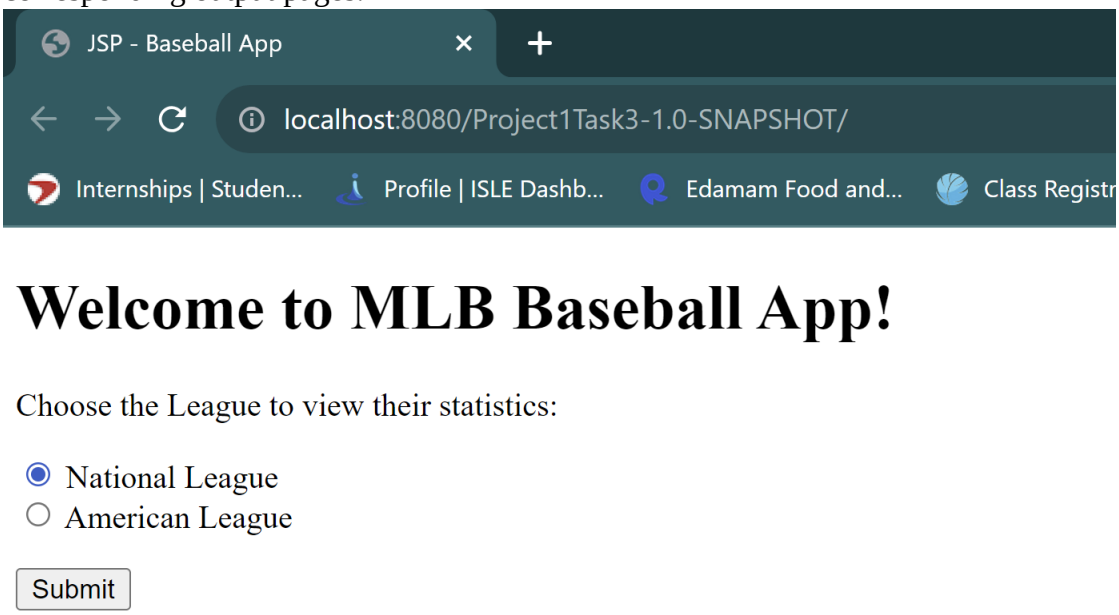


Fig 1: Input Page 1 – Choosing the League Option

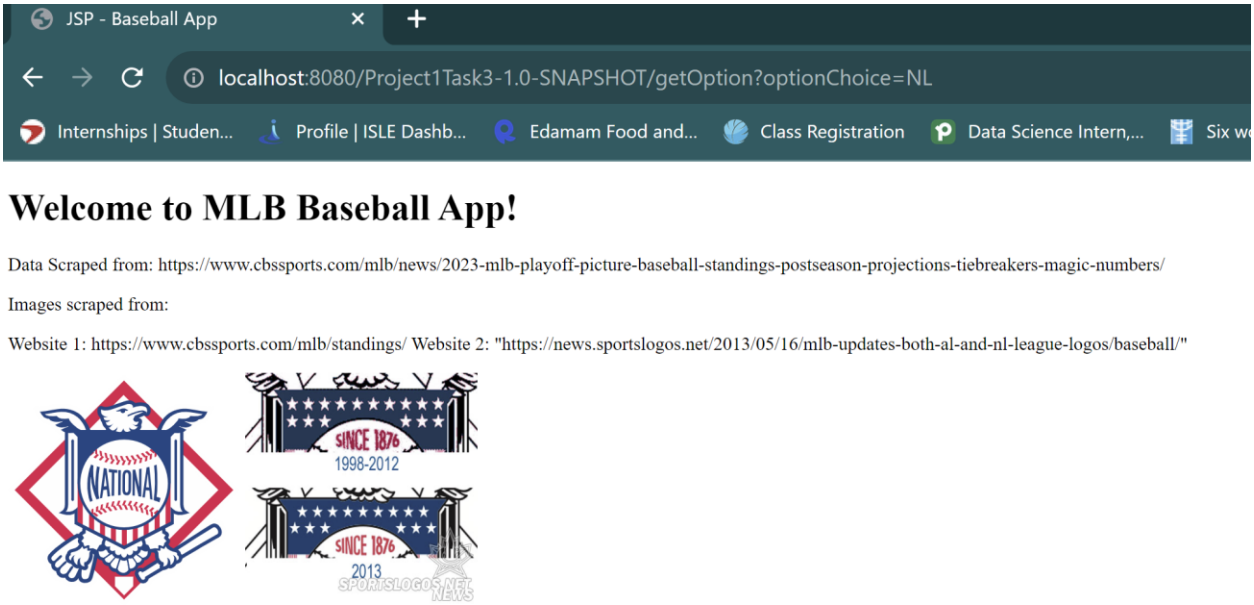
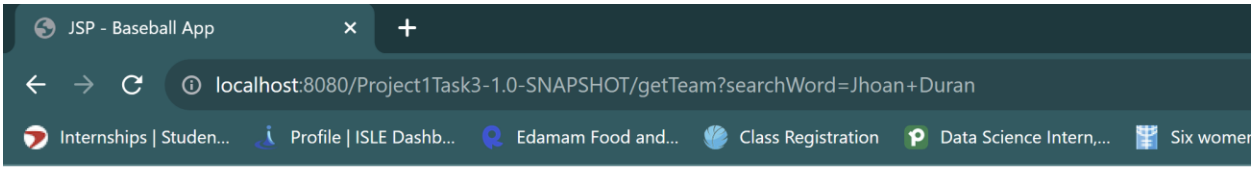


Fig 2: Ouput Page 1 – Displaying the corresponding leagues' logo and team statistics
Input Page 2: Search for a player to view their stats



Welcome to MLB Baseball App!

Scraped from "https://legacy.baseballprospectus.com/pitchfx/leaderboards/"

Here are the player statistics

Player Tm Th Num Velo H Mov V Mov Sw Rate Whf/Sw Foul/Sw GB/FB GB% LD% FB% PU% CS/Tk CS%

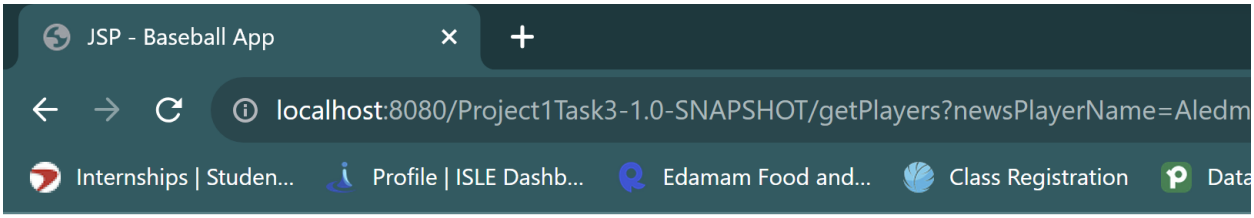
Jhoan Duran MIN R 425 101.87 -6.22 7.25 57.41% 36.07% 43.03% 2.27 53% 19% 23% 4% 26% 48%

Search for a player to view their recent news

Aledmys Diaz

Submit

Fig 3: Output Page 2: Displaying the player's stats
Input Page 3: Search for a player to view their recent news



Welcome to MLB Baseball App!

API CALL: https://appac.github.io/mlb-data-api-docs/

Player Name:

Aledmys Diaz

Player Related Recent News:

St. Louis Cardinals traded SS Aledmys Diaz to Toronto Blue Jays for CF J.B. Woodman.

Fig 4: Output Page 3: Displaying Player related news and their name

b) Code snippets from the Java code that screen scrapes, queries the API, and produces output.

Servlet:

```
//Reference for the doGet function: Interesting Picture: Lab 2
protected void doGet(HttpServletRequest request,
                     HttpServletResponse response)
        throws ServletException, IOException {

    // get the option choice - American League or National League
    String search = request.getParameter("optionChoice");

    //get the player name whose stats are to be fetched
    String player = request.getParameter("searchWord");

    //get the player name whose news is to be fetched
    String newsPlayerName = request.getParameter("newsPlayerName");

    // determine what type of device our user is
    String ua = request.getHeader("User-Agent");

    boolean mobile;
    // prepare the appropriate DOCTYPE for the view pages
    if (ua != null && ((ua.indexOf("Android") != -1) || (ua.indexOf("iPhone")
!= -1))) {
        mobile = true;
        /*
         * This is the latest XHTML Mobile doctype. To see the difference it
         * makes, comment it out so that a default desktop doctype is used
         * and view on an Android or iPhone.
         */
        request.setAttribute("doctype", "<!DOCTYPE html PUBLIC \"-
//WAPFORUM//DTD XHTML Mobile 1.2//EN\"
\"http://www.openmobilealliance.org/tech/DTD/xhtml-mobile12.dtd\">");
    } else {
        mobile = false;
        request.setAttribute("doctype", "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD
HTML 4.01 Transitional//EN\" \"http://www.w3.org/TR/html4/loose.dtd\">");
    }

    String nextView;

    //Check if the option choice parameter is present.

    if (search != null ) {
        String picSize = (mobile) ? "mobile" : "desktop";
        // use model with the league name input and choose the result view
        String teamStats = bam.scrapeMLBPTeamStats(search);
        /*
        Here the teamStats which is returned from the scrapeMLBTeamStats
function, is
        set as attribute teamStats which is displayed on page2.jsp
         */

        request.setAttribute("teamStats",teamStats);
```

```
        // Forward the request to the next view.
        nextView = "page2.jsp";
        RequestDispatcher view = request.getRequestDispatcher(nextView);
        view.forward(request, response);
        return;
    }
    // Transfer control over the correct "view"

    /*
    Check if the searched player is not null. If no results are found, then
    display No Results found on the result page.
     */
    if(player!=null)
    {
        //Call the function bam.scrapeMLBPlayerStats to fetch the searched
player's stats

        String playerStats = bam.scrapeMLBPlayerStats(player);
        System.out.println(playerStats);
         /*
        Here the teamStats which is returned from the scrapeMLBPlayerStats
function, is
        set as attribute playerStats which is displayed on page3.jsp
         */

        request.setAttribute("playerStats",playerStats);
        nextView = "page3.jsp";
        // Forward the request to the next view.
        RequestDispatcher view = request.getRequestDispatcher(nextView);
        view.forward(request, response);
        return;
    }

    /*
    Check if the searched player is not null. If no results are found, then
    display No Results found on the result page.
     */
    if(newsPlayerName!= null)
    {
        //Call the function bam.scrapeMLBPlayerStats to fetch the searched
player's recent news

        String newsData =  bam.getNewsData(newsPlayerName);

        /*
        Here the newsData which is returned from the scrapeMLBPlayerStats
function, is
        set as attribute newsData which is displayed on page4.jsp
         */
        request.setAttribute("newsPlayerName",newsPlayerName);
        request.setAttribute("newsData",newsData);

        // Forward the request to the next view.
        nextView = "page4.jsp";
        RequestDispatcher view = request.getRequestDispatcher(nextView);
        view.forward(request, response);
```

```
    }
    else {
        //goes back to the index.jsp if no results found at any stage
        System.out.println("No Results");
        nextView = "index.jsp";
        RequestDispatcher view = request.getRequestDispatcher(nextView);
        view.forward(request, response);
    }
}
```

Model:

```
public class BaseballAppModel {
    //function to scrape the TeamStats from
https://www.cbssports.com/mlb/news/2023-mlb-playoff-picture-baseball-
standings-postseason-projections-tiebreakers-magic-numbers/
    public String scrapeMLBPTeamStats(String leagueName) throws IOException {
        String result = null;
        Document doc =
Jsoup.connect("https://www.cbssports.com/mlb/news/2023-mlb-playoff-picture-
baseball-standings-postseason-projections-tiebreakers-magic-numbers/").get();


        // Create an HTML StringBuilder to build the table
        StringBuilder tableHtml = new StringBuilder();
        /*Syntax Reference: ChatGPT Prompt: How to build a table html from
web scraped data
         which can be directly output to the jsp

         */

        String finalHtml = null;

        //for scraping the 3 tables for every league
        Element table = null;
        Element table1 = null;
        Element table2 = null;

        System.out.println(leagueName);

        //for web scraping the image logos
        /*
        Website 1: https://www.cbssports.com/mlb/standings/
        Website 2: "https://news.sportslogos.net/2013/05/16/mlb-updates-both-
al-and-nl-league-logos/baseball/"
         */

        //functions to fetch the logos
        String logoURL = fetchLeagueLogo(leagueName);
        String logoURL2 = fetchLeagueLogo2(leagueName);


        //output the respective tables - based on the league name AL or NL
        if (leagueName.equalsIgnoreCase("AL")) {
            table = doc.select("table").get(0);
```

```
            table1 = doc.select("table").get(1);
            table2 = doc.select("table").get(2);

        } else if (leagueName.equalsIgnoreCase("NL")) {
            table = doc.select("table").get(3);
            table1 = doc.select("table").get(4);
            table2 = doc.select("table").get(5);

            // handling the edge case of no results found
        } else {
            result = "not found";
        }
        //appending the tables to the tableHtml StringBuilder
        tableHtml.append("<table>");
        //Looping through the tables of the website and fetching the rows
        Elements rows = table.select("tr");
        for (Element row : rows) {
            Elements cells = row.select("td,th");
            tableHtml.append("<tr>");
            for (Element cell : cells) {
                tableHtml.append("<td>").append(cell.text()).append("</td>");
            }
            tableHtml.append("</tr>");
        }
        tableHtml.append("<p></p>");
        Elements rows1 = table1.select("tr");
        for (Element row : rows1) {
            Elements cells = row.select("td,th");
            tableHtml.append("<tr>");
            for (Element cell : cells) {
                tableHtml.append("<td>").append(cell.text()).append("</td>");
            }
            tableHtml.append("</tr>");
        }
        tableHtml.append("<p></p>");

        Elements rows2 = table2.select("tr");
        for (Element row : rows2) {
            Elements cells = row.select("td,th");
            tableHtml.append("<tr>");
            for (Element cell : cells) {
                tableHtml.append("<td>").append(cell.text()).append("</td>");
            }
            tableHtml.append("</tr>");
        }
        tableHtml.append("</table>");

        // Generate the final HTML page with the table
        finalHtml = "<html><head></head><body> <img src= \" " + logoURL +" \"
width=\"200\" height=\"200\"</img> <img src= \"  "+ logoURL2 +" \"
width=\"200\" height=\"200\"</img>" + tableHtml.toString() +"
</body></html>";
        //pretty printing the table and returning the html to the servlet
        Document finalDoc = Jsoup.parse(finalHtml);
        OutputSettings settings = new OutputSettings();
        settings.prettyPrint(true);
        finalDoc.outputSettings(settings);
```

```
        return finalDoc.html();
    }

    //function to fetch the first logo
    //Reference: https://www.w3schools.com/html/html_images.asp
    private String fetchLeagueLogo(String leagueName) throws IOException {
        Document doc_cbs =
Jsoup.connect("https://www.cbssports.com/mlb/standings/").get();
        //Fetching all the image tags
        Elements images_cbs = doc_cbs.select("img");
        Element image_cbs = null;
        //Looping through all the images and finding the one which conatins
the league name in the URL
        for(Element i:images_cbs)
        {
            String a = i.attr("data-lazy");
            if(a.contains(leagueName))
            {
                image_cbs = i;
                //once found, ending the loop
                break;
            }

        }


        //returning the image URL
        return image_cbs.attr("data-lazy");
    }
    //function to fetch the second logo
    //Reference: https://www.w3schools.com/html/html_images.asp
    private String fetchLeagueLogo2(String leagueName) throws IOException {

        Document doc_mlb =
Jsoup.connect("https://news.sportslogos.net/2013/05/16/mlb-updates-both-al-
and-nl-league-logos/baseball/").get();
        //Fetching all the image tags
        Elements images_mlb = doc_mlb.select("img");

        Element image_mlb = null;
        //Looping through all the images and finding the one which conatins
the league name in the URL
        for(Element i:images_mlb)
        {
            String a = i.attr("src");
            //finding the image which contains the correct substring
            if(leagueName.equalsIgnoreCase("AL"))
            {
                if(a.contains(leagueName) && !(a.contains("NL")))
                {
                    image_mlb = i;
                    break;
                }
            }
            else if(leagueName.equalsIgnoreCase("NL"))
            {
                if(a.contains(leagueName) && !(a.contains("AL")))
```

```
                {
                    image_mlb = i;
                    //once found, ending the loop
                    break;
                }
            }


        }


        //returning the image URL
        return image_mlb.attr("src");
    }

    //function to scrape the player statistics from
https://legacy.baseballprospectus.com/pitchfx/leaderboards/
    //Reference: ChatGPT Prompt: Example code to scrape table data and form a
result string from the row using Jsoup
    public String scrapeMLBPlayerStats(String player) throws IOException {
        Document doc =
Jsoup.connect("https://legacy.baseballprospectus.com/pitchfx/leaderboards/").
get();

        // Initialize a variable to store the result stats
        String resultStats = "";

        // Find the table containing player stats (you may need to inspect
the HTML to find the correct table)
        Element table = doc.select("table").get(0);

        // Check if the table exists
        if (table != null) {
            // Iterate through rows in the table (skip the header row)
            Elements rows = table.select("tr");

            for (int i = 3; i < rows.size(); i++) {
                Element row = rows.get(i);

                // Extract the player name from the second cell (assuming
it's the second column)
                String playerName = row.select("td").get(1).text(); // Use
.get(1) to access the second <td> element

                // Check if the player name matches the desired player
                if (playerName.equalsIgnoreCase(player)) {
                    // Extract the entire row as stats
                    resultStats = row.text();

                    break; // Exit the loop once the player's row is found
                }
            }
        } else {
            //handling the case of no results found
            resultStats = "Table not found on the webpage.";
        }
        //returning the stats to the servlet
```

```
        return resultStats;

    }

    //Function for the API call
    /*
    This API finds the news for all players from a particular start date to
an end date.
    Here, given any player input, it iterates through the JSON and fetches
the news related to
    that particular player
     */
    //API: http://lookup-service-prod.mlb.com/
    //Function Reference: ChatGPT prompt- using JSON Array in Gson to iterate
over JSON fields
    public String getNewsData(String playerName) throws IOException {
        StringBuilder response = null;
        String note=null;
        // Build the URL with URI parameters
        URL url = new URL("http://lookup-service-
prod.mlb.com/json/named.transaction_all.bam?sport_code='mlb'&start_date='2017
1201'&end_date='20171231'");
        // Create a HttpURLConnection and set the request method to GET
        HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
        connection.setRequestMethod("GET");

        // Set request headers (if needed)
        connection.setRequestProperty("Content-Type", "application/json");

        // Get the response code
        int responseCode = connection.getResponseCode();

        // Check if the request was successful (HTTP status 200)
        if (responseCode == HttpURLConnection.HTTP_OK) {
            // Create a BufferedReader to read the response
            BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
            response = new StringBuilder();
            String line;

            // Read the response line by line
            while ((line = reader.readLine()) != null) {
                response.append(line);
            }

            // Close the reader and the connection
            reader.close();
            connection.disconnect();

            //creating the Gson library object
            Gson gson = new Gson();

            //preprocessing to clean up the Json
            JsonObject transaction = gson.fromJson(response.toString(),
JsonObject.class);
            JsonObject queryResults =
```

```
gson.fromJson(transaction.get("transaction_all").toString(),
JsonObject.class);
            JsonObject query =
gson.fromJson(queryResults.get("queryResults").toString(), JsonObject.class);
            JsonArray rowResults = query.get("row").getAsJsonArray();

            //iterating over the JsonArray to fetch the player's name and the
news
            for (int i = 0; i < rowResults.size(); i++) {
                String player =
String.valueOf(rowResults.get(i).getAsJsonObject().get("player"));
                player = player.replace("\"", "");

                //matching the input player with the player in the JSON
                if (player.equalsIgnoreCase(playerName)) {
                    note =
String.valueOf((rowResults.get(i).getAsJsonObject().get("note")));
                    note = note.replace("\"", "");

                }

            }


        }

            //returning the note to the servlet
            return note;
        }
    }
```

index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>JSP - Baseball App</title>
</head>
<body>
<h1><%= "Welcome to MLB Baseball App!" %>
</h1>

<p> Choose the League to view their statistics: </p>
<form action="getOption" method="GET">

    <input type="radio" id="NL" name="optionChoice" value="NL">
    <label for="NL">National League</label><br>
    <input type="radio" id="AL" name="optionChoice" value="AL">
    <label for="AL">American League</label><br>
    <p></p>

    <input type="submit"  value="Submit">
    <!Reference: https://www.w3schools.com/html/html_forms.asp>
    <!Reference: Interesting Picture - Lab 2 (result.jsp)>
```

```
</form>


<br/>
</body>
</html>
```

Page2.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>JSP - Baseball App</title>
</head>
<body>
<h1><%= "Welcome to MLB Baseball App!" %>
</h1>


<form action="getTeam" method="GET">

    <% if (request.getAttribute("teamStats") != null) { %>
    <p> Data Scraped from: https://www.cbssports.com/mlb/news/2023-mlb-
playoff-picture-baseball-standings-postseason-projections-tiebreakers-magic-
numbers/</p>
    <p> Images scraped from: </p>
    <p> Website 1: https://www.cbssports.com/mlb/standings/
        Website 2: "https://news.sportslogos.net/2013/05/16/mlb-updates-both-
al-and-nl-league-logos/baseball/"</p>
    <p> <%= request.getAttribute("teamStats") %> </p>
    <% } %>
    <p></p>
    <p>Search for a player to view their stats</p>
    <p></p>
    <input type="text" name="searchWord" value="" /><br>
    <p></p>
    <input type="submit" value="Submit" />
</form>
<!Reference: https://www.w3schools.com/html/html_forms.asp>
<!Reference: Interesting Picture - Lab 2 (result.jsp)>
<br/>

</body>
</html>
```

Page3.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
```

```
    <title>JSP - Baseball App</title>
</head>
<body>
<h1><%= "Welcome to MLB Baseball App!" %>
</h1>

<form action="getPlayers" method="GET">
    <% if (request.getAttribute("playerStats") != null) { %>
    <p> Scraped from
"https://legacy.baseballprospectus.com/pitchfx/leaderboards/"</p>
    <p><b> Here are the player statistics </b></p>
    <p><b>Player               Tm Th Num Velo H Mov V Mov Sw Rate Whf/Sw
Foul/Sw GB/FB GB% LD% FB% PU% CS/Tk CS%</b></p>
    <p><%= request.getAttribute("playerStats") %></p>
    <% } else { %>
    <p>No results found</p>
    <% } %>
    <p> <b> Search for a player to view their recent news </b></p>
    <input type="text" name="newsPlayerName" value="" /><br>
    <p></p>
    <input type="submit" value="Submit" />

    <!Reference: https://www.w3schools.com/html/html_forms.asp>
    <!Reference: Interesting Picture - Lab 2 (result.jsp)>
</form>
</body>
</html>
```

Page4.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>

    <title>JSP - Baseball App</title>
</head>
<body>
<h1><%= "Welcome to MLB Baseball App!" %>
</h1>
<form action="getPlayers" method="GET">
    <% if (request.getAttribute("newsPlayerName") != null) { %>
    <p> API CALL: https://appac.github.io/mlb-data-api-docs/ </p>

<!/json/named.transaction_all.bam?sport_code='mlb'&start_date={start_date}&en
d_date={end_date}>
    <p> </p>

    <% if (request.getAttribute("newsData") != null) { %>
    <p> <b>Player Name:</b> </p>
    <p> <%= request.getAttribute("newsPlayerName") %> </p>
    <p> <b> Player Related Recent News: </b></p>
    <p> <%= request.getAttribute("newsData") %> </p>
    <% } %>
    <% } %>
```

```
    <!Reference: https://www.w3schools.com/html/html_forms.asp>
    <!Reference: Interesting Picture - Lab 2 (result.jsp)>
</form>
</body>
</html>
```